

AD-A074 937

AIR FORCE MANPOWER AND PERSONNEL CENTER RANDOLPH AFB TX
HOW TO SPEED UP YOUR REGRESSION MODEL.(U)
MAY 79 T M BEATTY, K P JAMES

F/G 12/1

HAIR ACETATE

/ OF /

AD
AO74937

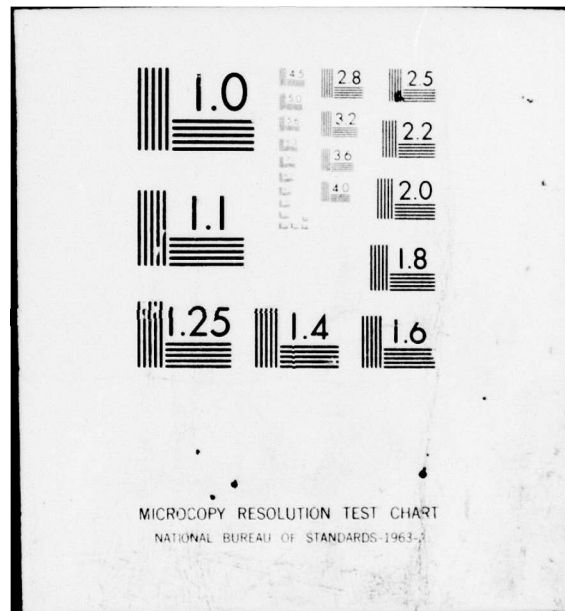
NL

END

DATE
FILMED

4-7X

DDC



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
6. How to Speed Up Your Regression Model.		9. Final rept.
10. T.M./BEATTY K.P./JAMES		5. PERFORMING ORG. REPORT NUMBER
7. PERFORMING ORGANIZATION NAME AND ADDRESS Directorate of Personnel Data Systems Air Force Manpower and Personnel Center Randolph AFB, Texas 78148		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS HQ Air Force Manpower and Personnel Center Randolph AFB, Texas 78148		12. REPORT DATE May 1979
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES 32 12 34
15. SECURITY CLASS. (of this report)		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)		Unclassified
Approved for public release, distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
regression multivariate regression least squares ordinary least squares		correlation intercorrelation matrix SPSS Biomedical Computer Programs
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Air Force personnel managers must be able to accurately forecast the force size. This need is explicit in meeting statutory budget limitations. Further, officer losses drive accessions, training, and promotion; thus the need for accuracy in forecasting losses cannot be over-emphasized. To accomplish this objective, loss rates have been generated using Ordinary Least Squares (OLS) stepwise regression run on what are locally dubbed the "binary files". The purpose of this paper is to report a front-end processor to OLS which has reduced computer run time by 85 percent for this organization.		

AD A U 7 4 9 3 7

DDC-FILE COPY

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

HOW TO SPEED UP YOUR REGRESSION MODEL

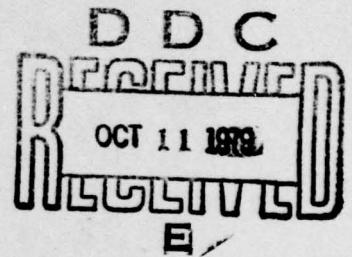
TECHNICAL MEMORANDUM

BY

T.M. BEATTY

K.P. JAMES

MAY 1979



**SYSTEMS SOFTWARE AND DEVELOPMENT BRANCH
SYSTEMS DEVELOPMENT AND SUPPORT DIVISION
DIRECTORATE OF PERSONNEL DATA SYSTEMS
AIR FORCE MANPOWER/PERSONNEL CENTER
RANDOLPH AFB, TEXAS 78148**

HOW TO SPEED UP YOUR REGRESSION MODEL
TECHNICAL MEMORANDUM

BY

T.M. BEATTY
K.P. JAMES

MAY 1979

APPROVED BY:



FREDERICK S. PHILLIPS, SR., COL, USAF
DIRECTOR, PERSONNEL DATA SYSTEMS

While the contents of this report are considered to be correct, they are subject to modification upon further study. This report does not promulgate official Air Force policies or positions. The technical conclusions are solely those of the authors.

FOREWORD

This report and the associated software were prepared by the Modeling Section of the Air Force Manpower and Personnel Center in response to the need to solve large multivariate regression problems (100 attributes with up to 30,000 observations) in the management of the one million plus personnel employed by the Air Force. The collaboration of the following people is acknowledged: LCDR C. Pennington, Lt D. Hemphill, CMS L. Staton, and TSG D. Francis.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

EXECUTIVE SUMMARY

Problem

Air Force Personnel managers must be able to accurately forecast the force size. Taken in a more general context, managers must be able to forecast a system response to independent stimuli. Ordinary Least Squares (O.L.S.) multivariate regression has been used to meet this need. O.L.S. Regression consumes inordinate ADP resources.

Objective

Reduce ADP resource usage in regression studies.

Approach

Isolate the portion(s) of regression which account for the greatest resource consumption and optimize.

Results

The major portion (75-95 percent) of the ADP resource was found to be expended in the computation of intercorrelation matrices. Since many regression problems are sparse due to the use of dummy variables, the introduction of logic to omit zero attribute values within an observation has provided ADP resource savings as high as 90 percent.

Conclusions

1. Regression problems of less than 91.5% density will yield economies.
2. Data files should not be processed with general utility regression programs.
3. As a second choice process data files with stand-alone correlation matrix builders.
4. The greatest economies will be realized if no data file is created for regression analysis. Rather insert correlation matrix build logic in the ADP system at the point where regression file(s) would be created.
5. Any of several matrix input regression packages may then be used to perform the actual stepwise or multiple regression.

ABSTRACT

Air Force personnel managers must be able to accurately forecast the force size. This need is explicit in meeting statutory budget limitations. Further, officer losses drive accessions, training, and promotion; thus the need for accuracy in forecasting losses cannot be over-emphasized. To accomplish this objective, loss rates have been generated using Ordinary Least Squares (OLS) stepwise regression, run on what are locally dubbed the "binary files". The purpose of this paper is to report a front-end processor to OLS which has reduced computer run time by 85 percent for this organization.

INTRODUCTION

Numerous software packages are available to solve the multivariate regression problem. To wit ; SPSS Incorporated's statistical package for the social sciences, the Biomedical Division's BMD02R, Burroughs' Advanced Statistical Inquiry System (BASIS), and Greenberger & Ward's Iterative Method. These methods and presumably a host of others capitalize upon the ability to sequentially consume virtually an unlimited amount of data, reduce the data to a correlation matrix, and "solve the problem". This feature of sequential processing of data into a relatively small core-resident matrix is one of the main selling points for OLS regression. However, the run time of most regression packages is notorious. In fact, the literature, both proprietary and public, note that providing a correlation matrix as input to the regression program will significantly reduce the regression run time. In application this approach saves time only if the user wishes to run various "sub-problems" against the same data file. (For example one may wish to make individual runs with the same independent variables against two or more dependent variables.) The lack of savings observed in running the single problem results because the data file must still be sequentially processed to build the correlation matrix. This leads to the incontrovertible conclusion that in order to make any money in reducing regression run time, the correlation matrix generation logic must be attacked. This, then is our approach.

This paper reports three methods which have resulted in significant computer resource savings in our application of O.L.S. It is presumed that readers of this paper are generally conversant with O.L.S. methodology. Therefore the objective of this paper will be to provide techniques which may provide reductions in run time, and not a tutorial on regression or correlation derivation.

METHOD

General

A straight-forward method of doing regression is to sequentially read the observations into an array, say VAR_i where i varies from 1 to the number of attributes and then execute code similar to the following:

$$SUM(I) = SUM(I) + VAR(I)$$

$$SSUM(I) = SSUM(I) + VAR(I) * VAR(I)$$

$$XYSUM(I,J) = YYSUM(I,J) + VAR(I) * VAR(J)$$

for $I = 1$ to number of attributes

$J = 1$ to number of attributes

After processing all observations as above, then compute the means, standard deviations, and the intercorellation matrix as below: Where N = the number of observations.

$$XMEAN(I) = SUM(I)/N$$

$$STDEV(I) = \left(\frac{SSUM(I) - N * XMEAN(I) * XMEAN(I)}{N-1} \right) ** 0.5$$

$$R(I,J) = \frac{XYSUM(I,J) - SUM(I) * SUM(J)/N}{(N-1) * STDEV(I) * STDEV(J)}$$

The above produces the means and standard deviations for the attributes and the Pearson Product-Moment correlation coefficients which can be used as input to a properly designed regression program. But at what cost?

If the above logic were applied to a 100 attribute problem with 3 thousand observations, the SUM (I) and SSUM (I) computations would each be executed 300 thousand times taking approximately 15 seconds CPU time. The XYSUM (I,J) computation would be executed 30 million times at a cost of about 11 hundred seconds CPU time. All of this is in addition to the time required to read the file and handle various other statements required to complete the set of executable program logic. (all timing estimates are for a Burroughs B6700).

There are obvious savings in the XYSUM (I,J) computations since the XYSUM matrix is symetric to the diagonal. Automatically the cost can be reduced by 49.5% or to 556 seconds of CPU time. Additionally the diagonal of the correlation matrix contains only 1's; therefore that XYSUM on the diagonal is extraneous. This reduces the time required to 545 seconds CPU. These economies are recognized by most, but not all, statistical packages which provide O.L.S. regression. There are, however, potential problems with the approach above.

Computers are limited in the number of significant digits that can be represented in a real number. And in regression we frequently deal with big numbers. Specifically, on the AFMPC B6700 a real number contains 11 significant digits. Thus truncation errors may develop if the sums, sums of squares, or cross-product sums exceed 10 to the 11th power. Depending upon cirmcumstances, this could result in attempting to compute the square root of a negative number or simply erroneous standard deviations and correlation elements. Since many regression

algorithms have not planned for this anomaly, the researcher is cautioned to consider this possibility when dealing with big numbers, particularly if the attempted regression run terminates with "INVALID ALOG ARGUMENT".

Thus far we've reduced the computation of XYSUM by over one-half for those statistical packages which do not recognize the symmetry of the matrix. The next area and the most important is the database itself. The data files we utilize daily are files created from raw data, the Master Personnel File (MPF). The raw data is converted to what we call "binary files". Each observation contains upwards of 100 variables or attributes. Most, but not all, attributes are "1" or "0", e.g., either the individual has a regular commission or he does not. Our new composite binary file contains on/off state variables and continuous variables, such as age. No statistical packages yet observed recognize the economics of checking for a value of "0" for an attribute before doing the SUM (I), SSUM (I), and XYSUM (I,J) computations. Our point is; a "0" added to a value of SUM (I) is the original SUM (I) and that squared, is the original SSUM (I), so why do them? By inclusion of 2 lines of code (in most cases), we check to see if the value is "0", increment the counter by one, and step back to process the next value instead of going thru all the calculations to end up with the same results. We use the same logic for XYSUM (I,J). The following is a sample of the code from our local regression package. The starred lines are the added code.


```

DO 80 J=1, KK
*IF(VAR(J).EQ.0) GO TO 80
SUM (J) = SUM (J) + VAR(J)
SSUM (J) = SSUM (J) + VAR (J) * VAR (J)

DO 110 K=1, J-1
*IF (VAR(K).EQ.0) GO TO 110
XYSUM (J,K)=XYSUM (J,K) + VAR (J) * VAR (K)
110 CONTINUE
80 CONTINUE

```

The added lines of code will cost us some CPU time. The overall CPU runtime reduction is clearly a function of the file density. A test problem was constructed to see at what point would the modifications to the regression break even with the regression run time. To accomplish this the inner and outer DO statements were timed in a variety of computer mixes on the B6700. To process a 10,000 observation problem with 100 variables; the average run time was:

outer computations	= 56 seconds
inner computations	= <u>2200 seconds</u>
Total	2256 seconds

The worst case with the new regression code and an 80% dense file was:

outer computations	45 seconds
inner computations	<u>2216 seconds</u>
Total	2261 seconds

If a file is 80% or less dense then the inclusion of the extra lines of code is cost effective.

To be able to reduce CPU time enables drastic reductions in clock time. Clock time represents a tie up of computer resources and personnel. Two actual files with 28% density are presented below:

- a. File size: 2374 observations
Attributes: 92 variables
File Density: 28%
File Format: Formatted
CPU run time: 1375 seconds versus 230 seconds
Clock time: 1 hr, 9 min, 57 sec versus 6 min 58 sec
- b. File size: 4197
Attributes: 94
File Density: 28%
File Format: Formatted
CPU run time: 2676 sec versus 322 sec.
Clock time: 2 hrs, 15 min 55 sec versus 10 min 46 sec

In example a., the CPU run time was reduced 83% while the clock time was reduced 90%. In example b. the CPU run time was reduced 87% while the clock time was reduced 96%. This is significant savings as the density of our average files is 28%. To restate the objectives we had; we needed to run regressions with matrix input and needed to speed up the matrix build process. The savings in CPU and clock time is geometric. Once a parent binary file is created capturing data at given points in time numerous regressions can be run on the file utilizing subsets of variables to answer a variety of questions in the same or less amount of time it took to run only one regression.

Application

First and foremost, the user must avoid processing the raw data with off-the-shelf, generalized software. And, if reading the data file can be avoided all together, even better. Locally we have modified production regression applications in both of the manners indicated.

In the first case our regression program was modified at the point where the data file would be sequentially consumed. At this point a procedure is invoked which processes all of the raw data and returns the intercorrelation matrix, the means, and the standard deviations to the regression program. Using the procedure shown below in figure #1, we have realized savings of 75-85 percent in CPU and in excess of 90 percent in elapsed time processing data files which average 28 percent density. This specific application has been generalized to produce a utility procedure which may be invoked by any calling program requiring the means, standard deviations, and a correlation matrix. Appendix A presents the utility procedure in ALGOL. As can be readily seen from the program documentation, this procedure may be included in a calling program to process observations as sequentially read, or introduced earlier in a system. It is this latter application which provides the greatest efficiency.

OBJECT/LOSS1/LODATA

```

$SET LINEINFO LIST
X PROCEDURE TO READ DATA FOR REGFUN
X
X
X
X
X INSERT INTO MAIN PROG : $SET AUTOBIND
X IF WIND=0
X CALL LOADDATA(IWIND,ISKIP,IREC,IPIRB,N,P,X,SIDEV,ZMEAN,A)
X
X
X
X
X
X [FILE FILE3,FILE6.]
X
X PROCEDURE LOADDATA(IWIND,ISKIP,IREC,IPIRB,N,P,X,SIDEV,ZMEAN,AA),
VALUE IWIND,ISKIP,N,IF,
INTEGER IWIND,ISKIP,IREC,IPIRB,N,IF,
ARRAY X(*), ZMEAN(*), AA(*), SIDEV(*)
BEGIN
    DEFINE BINSIZE=132#;
    FILE FILE3A(KIND=DISKPACK, PACKNAME='PRODATA.', FILETYPE=7,
        TITLE='BIN/LOSS75.'),
    FILE FILE3B(KIND=DISKPACK, PACKNAME='PRODATA.', FILETYPE=7,
        TITLE='BIN/LOSS76.'),
    FILE FILE3C(KIND=PRINTK);
    INTEGER I,T,J,K,L,N,C3,XI,CNT,
    REAL T1,T2,
    EBCDIC ARRAY BREGDIO(BINSIZE-1),
    ARRAY TAGLO(0),SUM,SUMTO(100),
    LABEL PAR.EOF.EXIT,
    DEFINE LDA(COL,LEN)=XI*(**1)+INTEGER(PRECDCOL),LEND#,
    A(I,J)=AA(I+99*J),
    YRGRP=(38,71#),
    FILNR=(31,21#),
    RECNR=(29,301#),
    T2:=TIME(12),
    IF NOT FILE3.OPEN THEN FILE3.OPEN.=TRUE,
    X
    X
    X IF IWIND.GE.O THEN BEGIN
        WRITE(FILE3), WRITE(FILE6, // "INPT FILE REVDIO "/>, END,
            // "SKIP",
            // "RECORDS SKIPPED "/>, // "SKIP",
            // "END.",
            // "END."
        )
    X
    X IF ISKIP.GT.O THEN BEGIN
        SPACE(FILE3,ISKIP), IREC:=+ISKIP,
        WRITE(FILE6, // "INPT FILE REVDIO "/>, // "SKIP",
            // "RECORDS SKIPPED "/>, // "SKIP",
            // "END.",
            // "END."
        )
    X

```

```

X
J=N-1; M=IP-1;
FOR I=0 STEP 1 UNTIL J DO BEGIN
  READ(FILE3, I, TAG, IEOF, PAR);
  IF TAG=1 THEN BEGIN
    THEN READ(FILE3, TAG, IEOF, PAR);
    ELSE READ(FILE3, TAG, IEOF, PAR);
  END;
  XI=-1;
  FOR K=16 STEP 1 UNTIL 24 DO BEGIN
    31 STEP 1 UNTIL 39, 129;
    45 STEP 1 UNTIL 57, 130, 63, 64;
    77 STEP 1 UNTIL 80;
  END;
  DO LDX(K, 1);
  IREC=0; IPROB=0;
  FOR K=0 STEP 1 UNTIL M DO BEGIN
    IF T=X(K) NEG 0 THEN BEGIN
      SUM(K) := X + T;
      SSUM(K) := X + T;
      FOR L=0 STEP 1 UNTIL K-1 DO
        IF X(L) NEG 0 THEN
          A(K, L) := X + T * X(L);
    END;
  END;
  END READ LOOP;
  C2 := CNT - 1;
  FOR K=0 STEP 1 UNTIL M DO BEGIN
    T := XMEAN(K) := SUM(K)/CNT;
    T1 := STDEV(K) := ((SUM(K)-CNT*T*T)/C2)*0.5;
    A(K, K) := 1;
    IF T1 NEG 0 THEN
      FOR L=0 STEP 1 UNTIL K-1 DO
        IF T := T1*STDEV(L) NEG 0 THEN
          A(K, L) := A(L, K) := ((A(K, L)-SUM(K)*SUM(L)/CNT)/C2)/T;
        ELSE A(K, L) := A(L, K) := 0.019;
      ELSE FOR L=0 STEP 1 UNTIL K-1 DO
        A(K, L) := A(L, K) := 0.019;
    END;
  END;
  T2 := TIME(12)-T2;
  WRITE(1, T2, CNT, M, K, L);
  GO EXIT;
X
X
PAR: WRITE(FILE6, "PARITY ERROR ON INPUT AT RECD NR ", 18,
  " THIS PROBLEM ", 19, IREC, IPROB);
X
MYSELF STATUS=-1;
X
X
EOF: WRITE(FILE6, "UNRECEIVED END OF INPUT AT RECD NR ", 18,
  " THIS PROBLEM ", 19, IREC, IPROB);
X
MYSELF STATUS=-1;
X
X

```

*Accumulate
data after each
iteration*

*Compute mean
standard
deviation, and
variance after
all data read*

EXIT: END OF LOAD DATA;

00000400 002 002 2 2

DATA IS 000F LONG

DATA IS 000A LONG

DATA(002) IS 010B LONG

*** WARNING : THIS PROCEDURE VALID FOR BINDING ONLY - FOR OTHER USE RECOMPILATE AT LEVEL 2 ***

BLOCK IS SEGMENT 0000A

BLOCK(004) IS 000C LONG

NUMBER OF ERRORS DETECTED = 0

NUMBER OF SEGMENTS = 9. TOTAL SEGMENT SIZE = 260 WORDS. CORE ESTIMATE = 1438 WORDS. STACK ESTIMATE = 45

PROGRAM SIZE = 107 CARDS, 792 SYNTACTIC ITEMS, 37 DISK SEGMENTS.

PROGRAM FILE NAME: OBJECT/LOSS1/LDATA

COMPILE TIME = 13.966 SECONDS ELAPSED; 2.727 SECONDS PROCESSING; 1.203 SECONDS I/O

Our second application of these enhancements to regression has been accomplished by early introduction into an existing system of software similar to that above, in figure #1. In this case a system in which two files were merged produced a "binary" file for subsequent processing in a stand-alone regression package. In the system as it previously operated, the two files were merged and written to diskpack for later regression analysis. At this point program logic was added to the "merge" program to compute the necessary data for producing the means, standard deviations, and correlation matrix. This, thus saves the ADP time to write and read a file plus the storage medium to store data between processing steps. Resource usage was markedly reduced.

The implementation in the second case realized the preatest reduction in computer resources. Regression problems which had run in 1400 CPU seconds are now processed in 120 CPU seconds. In addition DASD storage can be reduced.

Appendix B presents the HOL source code illustrating one way to insert the time consuming portion of regression into a extant ADP application.

CONCLUSION/RECOMMENDATION

- OLS regression requires high CPU resources.
- Applied regression typically produces low density files.
That is, introduction of dummy variables to accomodate non-linearity results in lots of zeros.
- Files of less than 80% density can be processed more rapidly by improved logic.
- Don't use off-the-shelf generalized regression packages to build correlation matrices.
- As a second choice use stand-alone correlation matrix generator.
- As a first choice hardwire correlation matrix logic into existing systems at the point where the data files for regression analyses are now produced.

4:41 PM FRIDAY, APRIL 27, 1979

```
0 $ SET ERLIST LINEINFO
100 * WARNING THIS PROCEDURE MUST BE BOUND I
200 *
300 * THIS PROCEDURE PRODUCES THE MEAN, STANDARD DEVIATION AND PEARSON'S
400 * PRODUCT-MOMENT INTERCORRELATION MATRIX FOR ANY NUMBER OF VARIABLES,
500 * AND ANY NUMBER OF OBSERVATIONS. THE MEAN, STANDARD DEVIATION AND
600 * INTERCORRELATION MATRIX ARE STORED IN A DISK-SACK FILE WITH A TITLE
700 * COMPOSED OF A DIRECTORY OR "MATRIX/DATA," AND A FILENAME SPECIFIED
800 * BY THE USER OR IT IS ABSENCE A FILENAME REFLECTING THE MONTH, DAY,
900 * YEAR, HOUR, MINUTE, AND SECOND THE FILE WAS CREATED. THE
1000 * FILENAME BEGINNING IN THE SECOND WORD OF THE ARRAY, B SHOULD
1100 * MINIMIZE THE IMPACT ON THE DEPENDENT VARIABLE OF TWO OR MORE HIGHLY
1200 * CORRELATED VARIABLES. THE HIGH CORRELATION THRESHOLD CAN BE SET
1300 * ANYWHERE FROM 0.0000 TO 1.0000 BASED ON A PARAMETER PASSED TO THE
1400 * PROCEDURE AT EXECUTION.
1500 * THE USER'S PROGRAM SHOULD CONTAIN A DECLARATION LIKE THIS:
1600 * PROCEDURE MATRIX(A,B); VALUE B; REAL B; ARRAY A(10); EXTERNAL;
1700 * FOR EACH OBSERVATION (I.E. RECORD READ), MATRIX SHOULD BE CALLED
1800 * PASSING THE VALUES OF THE VARIABLES IN THE ARRAY AND THE NUMBER OF
1900 * VARIABLES (I.E. THE SIZE OF THE ARRAY) ONE RELATIVE IN B. AFTER THE
2000 * LAST OBSERVATION, MATRIX SHOULD BE CALLED ONE MORE TIME WITH THE HIGH
2100 * CORRELATION THRESHOLD IN THE FIRST WORD OF THE ARRAY AND THE
2200 * FILENAME BEGINNING IN THE SECOND WORD OF THE ARRAY, B SHOULD
2300 * CONTAIN THE NUMBER OF THE DEPENDENT VARIABLE ZERO-RELATIVE.
2400 * ONE RESTRICTION IS THAT THE NUMBER OF VARIABLES WILL REMAIN THE
2500 * SAME FOR EVERY OBSERVATION IN EACH GROUP OR SET OF OBSERVATIONS.
2600 * I.E., EACH RECORD IN A FILE CONTAINS THE SAME NUMBER OF VARIABLES.
2700 * THE FIRST TIME AND EACH SUBSEQUENT TIME THRU THE PROCEDURE WHILE B
2800 * REMAINS THE SAME THE USER IS ACCUMULATING DATA. WHEN B CHANGES IT
2900 * INDICATES THAT THE USER WANTS TO FINISH THE CALCULATIONS AND
3000 * PRODUCE THE FILE CONTAINING THE MEAN, STANDARD DEVIATION AND THE
3100 * INTERCORRELATION MATRIX.
3200 * THE FILE IS CREATED AS A FRONT END TO SPSS AND THEREFORE IS IN A
3300 * FORMAT WHICH IT IS EXPECTING (I.E. EACH RECORD IS A CARD IMAGE THE
3400 * FIRST SET OF RECORDS CONTAINS THE MEANS OF THE VARIABLES, THE SECOND
3500 * SET OF RECORDS CONTAINS THE STANDARD DEVIATIONS OF THE VARIABLES AND
3600 * THE REST OF THE RECORDS CONTAIN THE INTERCORRELATION MATRIX. THE DATA
3700 * IS WRITTEN WITH AN F10.5 FORMAT EIGHT WORDS PER RECORD. THUS IF THERE
3800 * ARE MORE THAN EIGHT VARIABLES THE MEANS AND STANDARD DEVIATIONS WILL
3900 * BE WRITTEN ON MORE THAN ONE RECORD, AND IF THERE ARE MORE THAN TWO
4000 * VARIABLES THE INTERCORRELATION MATRIX WILL ALSO BE WRITTEN ON MORE
4100 * THAN ONE RECORD. THIS IS USER INFORMATION ONLY.)
4200 * IN ORDER TO BIND THIS ROUTINE YOU MUST FIRST CREATE A COPY OF THE
4300 * OBJECT CODE UTILITY/OR/MATRIX UNDER YOUR DIRECTORY. ADDITIONALLY YOU
4400 * MUST SET AUTOBIND IN YOUR SOURCE CODE. FOR MORE COMPLETE INSTRUCTIONS
4500 * READ EITHER GREGORY VOLUME II OR BURROUGHS BINDER MANUAL.
4600 * THIS IS A PRODUCT OF THE MODELING AND RESEARCH SECTION OF THE SYSTEM
4700 * SOFTWARE AND DEVELOPMENT BRANCH AT THE AIR FORCE MANPOWER AND
4800 * PERSONNEL CENTER (AFMPC/NRCPCH). THE ORIGINAL CONCEPT WAS DEVELOPED
4900 * BY MASTER THOMAS BEATTY, CHIEF MODELING AND RESEARCH SECTION. THE
5000 * PROCEDURE WAS DESIGNED AND WRITTEN BY FIRST LIEUTENANT DAVID HEMPHILL,
5100 * USAF. THE DESIGN AND PROGRAMMING EFFORT WAS ACCOMPLISHED OVER A PERIOD
5200 * OF NOVEMBER 1978 THRU APRIL 1979. ANY INQUIRIES MAY BE DIRECTED TO
5300 * EITHER MR BEATTY OR LT HEMPHILL AT AFMPC/NRCPCH7, RANDOLPH AFB TEX
5400 * 78148 OR AT (COMM) (512) 652- OR AUTO -487 & EXT 2233 OR 5694.)
5500 * PROCEDURE MATRIX(VARIABLES,REAL NUMB(RVARIABLES)).
5600 * VALUE NUMB(RVARIABLES),REAL NUMB(RVARIABLES).
```

```

5700      ARRAY VARIABLES ARRAY(1);
5800      BEGIN
5900      OWN ARRAY XYSUM(1:1, SUM, SSUM(1:1, 0));
6000      OWN BOOLEAN NOTFIRST;
6100      OWN REAL ARRAY SIZE, OBSERVATIONS;
6200      REAL I1, I2, T1, T2, T3;
6300      IF NOTFIRST THEN
6400      ELSE
6500      BEGIN
6600      NOTFIRST := TRUE;
6700      ARRAY SIZE := NUMBER VARIABLES;
6800      RESIZE(XYSUM(1, 1), ARRAY SIZE * ARRAY SIZE);
6900      RESIZE(SUM(1, 1), ARRAY SIZE);
7000      RESIZE(SSUM(1, 1), ARRAY SIZE);
7100      OBSERVATIONS := 0;
7200      END;
7300      IF ARRAY SIZE = NUMBER VARIABLES THEN
7400      BEGIN
7500      OBSERVATIONS := 1;
7600      FOR I1 := 0 STEP 1 UNTIL ARRAY SIZE - 1 DO
7700      IF I1 = VARIABLES ARRAY(1) THEN
7800      BEGIN
7900      SUM(I1) := 1;
8000      SSUM(I1) := 1;
8100      FOR I2 := 0 STEP 1 UNTIL I1 - 1 DO
8200      IF I2 = VARIABLES ARRAY(1) THEN
8300      XYSUM(I1, 1, 1) := 1;
8400      END;
8500      END;
8600      END;
8700      ELSE
8800      BEGIN
8900      THE PROCEDURE IS CALLED ONE MORE TIME AFTER THE LAST OBSERVATION AND
9000      THIS PIECE OF CODE IS EXECUTED TO REBUILD THE FILE CONTAINING THE
9100      MEAN, STANDARD DEVIATION, AND PEARSONS INTERCORRELATION MATRIX IN
9200      ADDITION HIGH CORRELATION VARIABLES ARE IDENTIFIED AND ELIMINATED
9300      THE EQUATION BASED ON A USER SUPPLIED TOLERANCE THE USER MAY
9400      SUPPLY A FILE NAME VIA THE REAL ARRAY BEGINNING IN THE SECOND WORD.
9500      THE TOLERANCE IS IN THE FIRST WORD AND THE DEPENDENT VARIABLE ZERO
9600      RELATIVE IS SUPPLIED IN THE SECOND WORD.
9700      FILE OUTPUT FILE NAME: PACK, MAXREC SIZE = 14, BLOCK SIZE = 40, AREASIZE = 120;
9800      FILE I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11, I12, I13, I14, I15, I16, I17, I18, I19, I20, I21, I22, I23, I24, I25, I26, I27, I28, I29, I30, I31, I32, I33, I34, I35, I36, I37, I38, I39, I40, I41, I42, I43, I44, I45, I46, I47, I48, I49, I50, I51, I52, I53, I54, I55, I56, I57, I58, I59, I60, I61, I62, I63, I64, I65, I66, I67, I68, I69, I70, I71, I72, I73, I74, I75, I76, I77, I78, I79, I80, I81, I82, I83, I84, I85, I86, I87, I88, I89, I90, I91, I92, I93, I94, I95, I96, I97, I98, I99, I100, I101, I102, I103, I104, I105, I106, I107, I108, I109, I110, I111, I112, I113, I114, I115, I116, I117, I118, I119, I120, I121, I122, I123, I124, I125, I126, I127, I128, I129, I130, I131, I132, I133, I134, I135, I136, I137, I138, I139, I140, I141, I142, I143, I144, I145, I146, I147, I148, I149, I150, I151, I152, I153, I154, I155, I156, I157, I158, I159, I160, I161, I162, I163, I164, I165, I166, I167, I168, I169, I170, I171, I172, I173, I174, I175, I176, I177, I178, I179, I180, I181, I182, I183, I184, I185, I186, I187, I188, I189, I190, I191, I192, I193, I194, I195, I196, I197, I198, I199, I200, I201, I202, I203, I204, I205, I206, I207, I208, I209, I210, I211, I212, I213, I214, I215, I216, I217, I218, I219, I220, I221, I222, I223, I224, I225, I226, I227, I228, I229, I230, I231, I232, I233, I234, I235, I236, I237, I238, I239, I240, I241, I242, I243, I244, I245, I246, I247, I248, I249, I250, I251, I252, I253, I254, I255, I256, I257, I258, I259, I260, I261, I262, I263, I264, I265, I266, I267, I268, I269, I270, I271, I272, I273, I274, I275, I276, I277, I278, I279, I280, I281, I282, I283, I284, I285, I286, I287, I288, I289, I290, I291, I292, I293, I294, I295, I296, I297, I298, I299, I300, I301, I302, I303, I304, I305, I306, I307, I308, I309, I310, I311, I312, I313, I314, I315, I316, I317, I318, I319, I320, I321, I322, I323, I324, I325, I326, I327, I328, I329, I330, I331, I332, I333, I334, I335, I336, I337, I338, I339, I340, I341, I342, I343, I344, I345, I346, I347, I348, I349, I350, I351, I352, I353, I354, I355, I356, I357, I358, I359, I360, I361, I362, I363, I364, I365, I366, I367, I368, I369, I370, I371, I372, I373, I374, I375, I376, I377, I378, I379, I380, I381, I382, I383, I384, I385, I386, I387, I388, I389, I390, I391, I392, I393, I394, I395, I396, I397, I398, I399, I400, I401, I402, I403, I404, I405, I406, I407, I408, I409, I410, I411, I412, I413, I414, I415, I416, I417, I418, I419, I420, I421, I422, I423, I424, I425, I426, I427, I428, I429, I430, I431, I432, I433, I434, I435, I436, I437, I438, I439, I440, I441, I442, I443, I444, I445, I446, I447, I448, I449, I450, I451, I452, I453, I454, I455, I456, I457, I458, I459, I460, I461, I462, I463, I464, I465, I466, I467, I468, I469, I470, I471, I472, I473, I474, I475, I476, I477, I478, I479, I480, I481, I482, I483, I484, I485, I486, I487, I488, I489, I490, I491, I492, I493, I494, I495, I496, I497, I498, I499, I500, I501, I502, I503, I504, I505, I506, I507, I508, I509, I510, I511, I512, I513, I514, I515, I516, I517, I518, I519, I520, I521, I522, I523, I524, I525, I526, I527, I528, I529, I530, I531, I532, I533, I534, I535, I536, I537, I538, I539, I540, I541, I542, I543, I544, I545, I546, I547, I548, I549, I550, I551, I552, I553, I554, I555, I556, I557, I558, I559, I560, I561, I562, I563, I564, I565, I566, I567, I568, I569, I570, I571, I572, I573, I574, I575, I576, I577, I578, I579, I580, I581, I582, I583, I584, I585, I586, I587, I588, I589, I590, I591, I592, I593, I594, I595, I596, I597, I598, I599, I600, I601, I602, I603, I604, I605, I606, I607, I608, I609, I610, I611, I612, I613, I614, I615, I616, I617, I618, I619, I620, I621, I622, I623, I624, I625, I626, I627, I628, I629, I630, I631, I632, I633, I634, I635, I636, I637, I638, I639, I640, I641, I642, I643, I644, I645, I646, I647, I648, I649, I650, I651, I652, I653, I654, I655, I656, I657, I658, I659, I660, I661, I662, I663, I664, I665, I666, I667, I668, I669, I670, I671, I672, I673, I674, I675, I676, I677, I678, I679, I680, I681, I682, I683, I684, I685, I686, I687, I688, I689, I690, I691, I692, I693, I694, I695, I696, I697, I698, I699, I700, I701, I702, I703, I704, I705, I706, I707, I708, I709, I710, I711, I712, I713, I714, I715, I716, I717, I718, I719, I720, I721, I722, I723, I724, I725, I726, I727, I728, I729, I730, I731, I732, I733, I734, I735, I736, I737, I738, I739, I740, I741, I742, I743, I744, I745, I746, I747, I748, I749, I750, I751, I752, I753, I754, I755, I756, I757, I758, I759, I760, I761, I762, I763, I764, I765, I766, I767, I768, I769, I770, I771, I772, I773, I774, I775, I776, I777, I778, I779, I780, I781, I782, I783, I784, I785, I786, I787, I788, I789, I790, I791, I792, I793, I794, I795, I796, I797, I798, I799, I800, I801, I802, I803, I804, I805, I806, I807, I808, I809, I810, I811, I812, I813, I814, I815, I816, I817, I818, I819, I820, I821, I822, I823, I824, I825, I826, I827, I828, I829, I830, I831, I832, I833, I834, I835, I836, I837, I838, I839, I840, I841, I842, I843, I844, I845, I846, I847, I848, I849, I850, I851, I852, I853, I854, I855, I856, I857, I858, I859, I860, I861, I862, I863, I864, I865, I866, I867, I868, I869, I870, I871, I872, I873, I874, I875, I876, I877, I878, I879, I880, I881, I882, I883, I884, I885, I886, I887, I888, I889, I890, I891, I892, I893, I894, I895, I896, I897, I898, I899, I900, I901, I902, I903, I904, I905, I906, I907, I908, I909, I910, I911, I912, I913, I914, I915, I916, I917, I918, I919, I920, I921, I922, I923, I924, I925, I926, I927, I928, I929, I930, I931, I932, I933, I934, I935, I936, I937, I938, I939, I940, I941, I942, I943, I944, I945, I946, I947, I948, I949, I950, I951, I952, I953, I954, I955, I956, I957, I958, I959, I960, I961, I962, I963, I964, I965, I966, I967, I968, I969, I970, I971, I972, I973, I974, I975, I976, I977, I978, I979, I980, I981, I982, I983, I984, I985, I986, I987, I988, I989, I990, I991, I992, I993, I994, I995, I996, I997, I998, I999, I1000, I1001, I1002, I1003, I1004, I1005, I1006, I1007, I1008, I1009, I1010, I1011, I1012, I1013, I1014, I1015, I1016, I1017, I1018, I1019, I1020, I1021, I1022, I1023, I1024, I1025, I1026, I1027, I1028, I1029, I1030, I1031, I1032, I1033, I1034, I1035, I1036, I1037, I1038, I1039, I1040, I1041, I1042, I1043, I1044, I1045, I1046, I1047, I1048, I1049, I1050, I1051, I1052, I1053, I1054, I1055, I1056, I1057, I1058, I1059, I1060, I1061, I1062, I1063, I1064, I1065, I1066, I1067, I1068, I1069, I1070, I1071, I1072, I1073, I1074, I1075, I1076, I1077, I1078, I1079, I1080, I1081, I1082, I1083, I1084, I1085, I1086, I1087, I1088, I1089, I1090, I1091, I1092, I1093, I1094, I1095, I1096, I1097, I1098, I1099, I1100, I1101, I1102, I1103, I1104, I1105, I1106, I1107, I1108, I1109, I1110, I1111, I1112, I1113, I1114, I1115, I1116, I1117, I1118, I1119, I1120, I1121, I1122, I1123, I1124, I1125, I1126, I1127, I1128, I1129, I1130, I1131, I1132, I1133, I1134, I1135, I1136, I1137, I1138, I1139, I1140, I1141, I1142, I1143, I1144, I1145, I1146, I1147, I1148, I1149, I1150, I1151, I1152, I1153, I1154, I1155, I1156, I1157, I1158, I1159, I1160, I1161, I1162, I1163, I1164, I1165, I1166, I1167, I1168, I1169, I1170, I1171, I1172, I1173, I1174, I1175, I1176, I1177, I1178, I1179, I1180, I1181, I1182, I1183, I1184, I1185, I1186, I1187, I1188, I1189, I1190, I1191, I1192, I1193, I1194, I1195, I1196, I1197, I1198, I1199, I1200, I1201, I1202, I1203, I1204, I1205, I1206, I1207, I1208, I1209, I1210, I1211, I1212, I1213, I1214, I1215, I1216, I1217, I1218, I1219, I1220, I1221, I1222, I1223, I1224, I1225, I1226, I1227, I1228, I1229, I1230, I1231, I1232, I1233, I1234, I1235, I1236, I1237, I1238, I1239, I1240, I1241, I1242, I1243, I1244, I1245, I1246, I1247, I1248, I1249, I1250, I1251, I1252, I1253, I1254, I1255, I1256, I1257, I1258, I1259, I1260, I1261, I1262, I1263, I1264, I1265, I1266, I1267, I1268, I1269, I1270, I1271, I1272, I1273, I1274, I1275, I1276, I1277, I1278, I1279, I1280, I1281, I1282, I1283, I1284, I1285, I1286, I1287, I1288, I1289, I1290, I1291, I1292, I1293, I1294, I1295, I1296, I1297, I1298, I1299, I1300, I1301, I1302, I1303, I1304, I1305, I1306, I1307, I1308, I1309, I1310, I1311, I1312, I1313, I1314, I1315, I1316, I1317, I1318, I1319, I1320, I1321, I1322, I1323, I1324, I1325, I1326, I1327, I1328, I1329, I1330, I1331, I1332, I1333, I1334, I1335, I1336, I1337, I1338, I1339, I1340, I1341, I1342, I1343, I1344, I1345, I1346, I1347, I1348, I1349, I1350, I1351, I1352, I1353, I1354, I1355, I1356, I1357, I1358, I1359, I1360, I1361, I1362, I1363, I1364, I1365, I1366, I1367, I1368, I1369, I1370, I1371, I1372, I1373, I1374, I1375, I1376, I1377, I1378, I1379, I1380, I1381, I1382, I1383, I1384, I1385, I1386, I1387, I1388, I1389, I1390, I1391, I1392, I1393, I1394, I1395, I1396, I1397, I1398, I1399, I1400, I1401, I1402, I1403, I1404, I1405, I1406, I1407, I1408, I1409, I1410, I1411, I1412, I1413, I1414, I1415, I1416, I1417, I1418, I1419, I1420, I1421, I1422, I1423, I1424, I1425, I1426, I1427, I1428, I1429, I1430, I1431, I1432, I1433, I1434, I1435, I1436, I1437, I1438, I1439, I1440, I1441, I1442, I1443, I1444, I1445, I1446, I1447, I1448, I1449, I1450, I1451, I1452, I1453, I1454, I1455, I1456, I1457, I1458, I1459, I1460, I1461, I1462, I1463, I1464, I1465, I1466, I1467, I1468, I1469, I1470, I1471, I1472, I1473, I1474, I1475, I1476, I1477, I1478, I1479, I1480, I1481, I1482, I1483, I1484, I1485, I1486, I1487, I1488, I1489, I1490, I1491, I1492, I1493, I1494, I1495, I1496, I1497, I1498, I1499, I1500, I1501, I1502, I1503, I1504, I1505, I1506, I1507, I1508, I1509, I1510, I1511, I1512, I1513, I1514, I1515, I1516, I1517, I1518, I1519, I1520, I1521, I1522, I1523, I1524, I1525, I1526, I1527, I1528, I1529, I1530, I1531, I1532, I1533, I1534, I1535, I1536, I1537, I1538, I1539, I1540, I1541, I1542, I1543, I1544, I1545, I1546, I1547, I1548, I1549, I1550, I1551, I1552, I1553, I1554, I1555, I1556, I1557, I1558, I1559, I1560, I1561, I1562, I1563, I1564, I1565, I1566, I1567, I1568, I1569, I1570, I1571, I1572, I1573, I1574, I1575, I1576, I1577, I1578, I1579, I1580, I1581, I1582, I1583, I1584, I1585, I1586, I1587, I1588, I1589, I1590, I1591, I1592, I1593, I1594, I1595, I1596, I1597, I1598, I1599, I1600, I1601, I1602, I1603, I1604, I1605, I1606, I1607, I1608, I1609, I1610, I1611, I1612, I1613, I1614, I1615, I1616, I1617, I1618, I1619, I1620, I1621, I1622, I1623, I1624, I1625, I1626, I1627, I1628, I1629, I1630, I1631, I1632, I1633, I1634, I1635, I1636, I1637, I1638, I1639, I1640, I1641, I1642, I1643, I1644, I1645, I1646, I1647, I1648, I1649, I1650, I1651, I1652, I1653, I1654, I1655, I1656, I1657, I1658, I1659, I1660, I1661, I1662, I1663, I1664, I1665, I1666, I1667, I1668, I1669, I1670, I1671, I1672, I1673, I1674, I1675, I1676, I1677, I1678, I1679, I1680, I1681, I1682, I1683, I1684, I1685, I1686, I1687, I1688, I1689, I1690, I1691, I1692, I1693, I1694, I1695, I1696, I1697, I1698, I1699, I1700, I1701, I1702, I1703, I1704, I1705, I1706, I1707, I1708, I1709, I1710, I1711, I1712, I1713, I1714, I1715, I1716, I1717, I1718, I1719, I1720, I1721, I1722, I1723, I1724, I1725, I1726, I1727, I1728, I1729, I1730, I1731, I1732, I1733, I1734, I1735, I1736, I1737, I1738, I1739, I1740, I1741, I1742, I1743, I1744, I1745, I1746, I1747, I1748, I1749, I1750, I1751, I1752, I1753, I1754, I1755, I1756, I1757, I1758, I1759, I1760, I1761, I1762, I1763, I1764, I1765, I1766, I1767, I1768, I1769, I1770, I1771, I1772, I1773, I1774, I1775, I1776, I1777, I1778, I1779, I1780, I1781, I1782, I1783, I1784, I1785, I1786, I1787, I1788, I1789, I1790, I1791, I1792, I1793, I1794, I1795, I1796, I1797, I1798, I1799, I1800, I1801, I1802, I1803, I1804, I1805, I1806, I1807, I1808, I1809, I1810, I1811, I1812, I1813, I1814, I1815, I1816, I1817, I1818, I1819, I1820, I1821, I1822, I1823, I1824, I1825, I1826, I1827, I1828, I1829, I1830, I1831, I1832, I1833, I1834, I1835, I1836, I1837, I1838, I1839, I1840, I1841, I1842, I1843, I1844, I1845, I1846, I1847, I1848, I1849, I1850, I1851, I1852, I1853, I1854, I1855, I1856, I1857, I1858, I1859, I1860, I1861, I1862, I1863, I1864, I1865, I1866, I1867, I1868, I1869, I1870, I1871, I1872, I1873, I1874, I1875, I1876, I1877, I1878, I1879, I1880, I1881, I1882, I1883, I1884, I1885, I1886, I1887, I1888, I1889, I1890, I1891, I1892, I1893, I1894, I1895, I1896, I1897, I1898, I1899, I1900, I1901, I1902, I1903, I1904, I1905, I1906, I1907, I1908, I1909, I1910, I1911, I1912, I1913, I1914, I1915, I1916, I1917, I1918, I1919, I1920, I1921, I1922, I1923, I1924, I1925, I1926, I1927, I1928, I1929, I1930, I1931, I1932, I1933, I1934, I1935, I1936, I1937, I1938, I1939, I1940, I1941, I1942, I1943, I1944, I1945, I1946, I1947, I1948, I1949, I1950, I1951, I1952, I1953, I1954, I1955, I1956, I1957, I1958, I1959, I1960, I1961, I1962, I1963, I1964, I1965, I1966, I1967, I1968, I1969, I1970, I1971, I1972, I1973, I1974, I1975, I1976, I1977, I1978, I1979, I1980, I1981, I1982, I1983, I1984, I1985, I1986, I1987, I1988, I1989, I1990, I1991, I1992, I1993, I1994, I1995, I1996, I1997, I1998, I1999, I2000, I2001, I2002, I2003, I2004, I2005, I2006, I2007, I2008, I2009, I2010, I2011, I2012, I2013, I2014, I2015, I2016, I2017, I2018, I2019, I2020, I2021, I2022, I2023, I2024, I2025, I2026, I2027, I2028, I2029, I2030, I2031, I2032, I2033, I2034, I2035, I2036, I2037, I2038, I2039, I2040, I2041, I2042, I2043, I2044, I2045, I2046, I2047, I2048, I2049, I2050, I2051, I2052, I2053, I2054, I2055, I2056, I2057, I2058, I2059, I2060, I2061, I2062, I2063, I2064, I2065, I2066, I2067, I2068, I2069, I2070, I2071, I2072, I2073, I2074, I2075, I2076, I2077, I2078, I2079, I2080, I2081, I2082, I2083, I2084, I2085, I2086, I2087, I2088, I2089, I2090, I2091, I2092, I2093, I2094, I2095, I2096, I2097, I2098, I2099, I2100, I2101, I2102, I2103, I2104, I2105, I2106, I2107, I2108, I2109, I2110, I2111, I2112, I2113, I2114, I2115, I2116, I2117, I2118, I2119, I2120, I2121, I2122, I2123, I2124, I2125, I2126, I2127, I2128, I2129, I2130, I2131, I2132, I2133, I2134, I2135, I213
```



```

10400      T1:=TIME(7);
10500      REPLAGE WORK BY "MATRIX/DATA/"; T1(29:61) FOR 2 DIGITS,
10600      MONI( (INTEGER(T1(35:61)-1)*3) FOR 3, T1 (47:12) FOR 2 DIGITS,
10700      "H", T1 (23:6) FOR 2 DIGITS, "M", T1 (17:6) FOR 2 DIGITS, "S",
10800      T1 (11:6) FOR 2 DIGITS, " ");
10900      END;
11000      REPLACE OUTFILE TITLE BY WORK;
11100      T3:=OBSERVATIONS-1;
11200      FOR I1:=0 STEP 1 UNTIL ARRAYSIZE-1 DO
11300          BEGIN
11400              T1:=MEAN(I1):=SUM(I1)/OBSERVATIONS;
11500              T2:=STANDARDDEVIATION(I1):=((SUM(I1)-OBSERVATIONS*T1*T1)/T3)
11600              *0.5;
11700              WRITEL( (LNGOUT, 4115, 2E10, 54, 11, 11, 12);
11800              XYSUM(I1)*ARRAYSIZE+I1):=1;
11900              IF T2 NEQ 0 THEN
12000                  FOR I2:=0 STEP 1 UNTIL I1-1 DO
12100                      IF T1:=12*STANDARDDEVIATION(I2) NEQ 0 THEN
12200                          BEGIN
12300                              T4:=XYSUM(I1)*ARRAYSIZE+I21:=XYSUM(I2)*ARRAYSIZE+I11):=
12400                              ((XYSUM(I1)*ARRAYSIZE+I2)-SUM(I1)*SUM(I2)/OBSERVATIONS)
12500                              /I2/I1;
12600                              IF ABS(T4) >= TOLERANCE AND I1 NEQ DEPENDENT AND I2 NEQ
12700                              DEPENDENT THEN
12800                                  BEGIN
12900                                      FOR I3:=0 STEP 1 UNTIL I5-1 DO
13000                                          IF HIT(I3) = 1 THEN
13100                                              HIT:=TRUE;
13200                                              IF NOT HIT THEN
13300                                                  BEGIN
13400                                                      HIT(I5):=I1;
13500                                                      IS:=*+1;
13600                                                  END;
13700                                                  HIT:=FALSE;
13800                                                  FOR I3:=0 STEP 1 UNTIL I5-1 DO
13900                                                      IF HIT(I3) = 12 THEN
14000                                                          HIT:=TRUE;
14100                                                          IF NOT HIT THEN
14200                                                              BEGIN
14300                                                                  HIT(I5):=I2;
14400                                                                  IS:=*+1;
14500                                                              END;
14600                                                                  HIT:=FALSE;
14700                                                                  END;
14800                                                                  ELSE
14900                                                                      XYSUM(I1)*ARRAYSIZE+I21:=XYSUM(I2)*ARRAYSIZE+I11):=0.0019
15000                                                                      FOR I2:=0 STEP 1 UNTIL I1-1 DO
15100                                                                          XYSUM(I1)*ARRAYSIZE+I21:=XYSUM(I2)*ARRAYSIZE+I11):=0.0019;
15200                                                                          END;
15300                                                                          FOR I1:=0 STEP 1 UNTIL I5-1 DO
15400                                                                              BEGIN
15500                                                                                  HIT(I11):=XYSUM(DEPENDENT*ARRAYSIZE+HIT(I11));
15600                                                                                  END;
15700                                                                                  FOR I1:=0 STEP 1 UNTIL I5-1 DO
15800                                                                                      T1:=ABS(HIT(I11));
15900                                                                                      BEGIN
16000                                                                                          FOR I2:=0 STEP 1 UNTIL I5-1 DO
16100                                                                                              IF T2:=ABS(HIT(I2)) > T1 THEN
16200

```



```

16300      BEGIN
16400      T1:=T2;
16500      I3:=I2;
16600      END;
16700      H12(I3):=0;
16800      FOR I2:=0 STEP 1 UNTIL I3-1, I3+1 STEP 1 UNTIL I5-1 DO
16900          IF ABS(XYSUMH1(I3)*ARRAYSIZE+H1(I2)) >= TOLERANCE THEN
17000              BEGIN
17100                  WRITE(OUTFILE, <16, " REMOVED DUE TO HIGH CORRELATION WITH ",
17200                      I45-H1(I2), H1(I3)),
17300                      FOR I4:=0 STEP 1 UNTIL H1(I2)-1, H1(I2)+1 STEP 1 UNTIL
17400                          ARRAYSIZE-1 DO
17500                              XYSUM(I1+ARRAYSIZE+H1(I2)):=XYSUMH1(I2)*ARRAYSIZE+I4);
17600                      END;
17700                  END;
17800                  WRITE(OUTFILE, <17, OBSERVATIONS);
17900                  WRITE(OUTFILE, <18, I0-5, MEAN);
18000                  WRITE(OUTFILE, <19, I0-5, STANDARDDEVIATION);
18100                  FOR I1:=0 STEP 1 UNTIL ARRAYSIZE-1 DO
18200                      WRITE(OUTFILE, <20, I0-5, FOR I2:=0 STEP 1 UNTIL ARRAYSIZE-1 DO
18300                          XYSUM(I1+ARRAYSIZE+I2);
18400                      LOCK(OUTFILE);
18500                      DEALLOCATE(XYSUM);
18600                      DEALLOCATE(SSUM);
18700                      DEALLOCATE(SSUM);
18800                      NOTFIRST:=FALSE;
18900                      END;
19000                  END;
19100      END;

```

```

00016300
00016400
00016500
00016600
00016700
00016800
00016900
00017000
00017100
00017200
00017300
00017400
00017500
00017600
00017700
00017800
00017900
00018000
00018100
00018200
00018300
00018400
00018500
00018600
00018700
00018800
00018900
00019000
00019100

```

7:48 AM MONDAY, APRIL 30, 1979

FRANCIS/SOURCE/NEWMERGE (04/23/79)

100 \$ RESET LIST 00000100
 200 \$ SET ERRLIST LINEINFO 00000200
 300 \$*****SGT FRANCIS OPHODA EXT-2233 00000300
 400 \$*****CARDIN FORMAT 00000400
 500 \$ CC 1-5 BOARD NAME 00000500
 600 \$ CC 1-2 CY 00000600
 700 \$ CC 3 TYPE BOARD 00000700
 800 \$ R-REG APPT 00000800
 900 \$ N-NOR 00000900
 1000 \$ P-PEM 00001000
 1100 \$ T-TEMP 00001100
 1200 \$ CC 4-5 GRADE BOARD OR TENURE OF APPT 00001200
 1300 \$ CC 70-72 TERT ZONE CUTOFF 00001300
 1400 \$ CC 74-76 SEC ZONE CUTOFF 00001400
 1500 \$ CC 78-80 PRI ZONE CUTOFF 00001500
 1600 BEGIN 00001600
 1700 FILE BINARY (KIND=DISKPACK, PACKNAME="MPCDATA", TITLE="BIN", MYUSE=10, 00001700
 1800 UNITS=CHARACTERS, FILETYPE=7); 00001800
 1900 FILE BOARDFILE (KIND=TAPE, BLOCKSIZE=6400, MAXRECSIZE=640, TITLE="BF", 00001900
 2000 UNITS=CHARACTERS, FILETYPE=7); 00002000
 2100 FILE LINE (KIND=PRINTER); 00002100
 2200 FILE CRD (KIND=READER, TITLE="CARDIN"); 00002200
 2300 FILE PRKIND=DISKPACK, PACKNAME="MPCDATA", MAXRECSIZE=14, 00002300
 2400 TITLE="TMB/MAT/106P", BLOCKSIZE=2100, PROTECTION=SAVE, AREAS=10, 00002400
 2500 AREASIZE=14); 00002500
 2600 SEC KIND=DISKPACK, PACKNAME="MPCDATA", MAXRECSIZE=14, 00002600
 2700 TITLE="TMB/MAT/106S", BLOCKSIZE=2100, PROTECTION=SAVE, AREAS=10, 00002700
 2800 AREASIZE=14); 00002800
 2900 TER KIND=DISKPACK, PACKNAME="MPCDATA", MAXRECSIZE=14, 00002900
 3000 TITLE="TMB/MAT/106T", BLOCKSIZE=2100, PROTECTION=SAVE, AREAS=10, 00003000
 3100 AREASIZE=14); 00003100
 3200 DIRECT ARRAY SUMS, SUMS, SUMP, SUMT, SUMTT(0:100); 00003200
 3300 DIRECT ARRAY XYSUMS, XYSUMP, XYSUMT(0:10000); 00003300
 3400 ARRAY RECHOLD, MEANT, MEANS, MEANP, STDVT, STDVS, STDVP, IAPRI(0:100); 00003400
 3500 ARRAY IASEC, IATERLO:1001; 00003500
 3600 EBCDIC ARRAY BIN(0:161); 00003600
 3700 BF(0:639); 00003700
 3800 ARRAY DISTLO:3, 0:100, 0:21, 00003800
 3900 CUISCORE(0:3); 00003900
 4000 ZNCNT(0:3); 00004000
 4100 ZONECNT(0:3); 00004100
 4200 STATDIST1(0:3, 0:200, 0:1); 00004200
 4300 STATDIST2(0:3, 0:1, 0:21); 00004300
 4400 BOOLEAN BOO: INTEGER PARCNT: LABEL PARR, 00004400
 4500 EBCDIC VALUE ARRAY TITL (" 00004500
 4600 "PRIMAR", "Y 00004600
 4700 "SECOND", "ARY 00004700
 4800 "TERTIA", "RY 00004800
 4900 FORMAT CARDINX2 A1, A3, X63, A3, X1, A3, X1, A3; 00004900
 5000 FORMAT HORIN("PREDICTED", X12, "PRIMARY ZONE"); 00005000
 5100 FORMAT HORIL("PREDICTED", X12, "PRIMARY ZONE", X13, "SECONDARY ZONE"); 00005100
 5200 FORMAT HORIR("PREDICTED", X12, "PRIMARY ZONE", X12, "SECONDARY ZONE", 00005200
 5300 X11, "TERTIARY ZONE"); 00005300
 5400 FORMAT HORZNI(" SCORE", X12, "TOTAL 00005400
 5500 SELECTED"); 00005500
 5600 FORMAT HORZIL(" SCORE", X12, "TOTAL 00005600
 5700 SELECTED", X10); 00005700
 5800 FORMAT HORZRI(" SCORE", X12, "TOTAL 00005800
 5900 SELECTED", X10); 00005900
 6000 FORMAT HOR3X14, "SELECT > CUT = CUT < CUT NON-SELECT > CUT", 00006000

Appendix B

NONWELL PAGE PRINTING SYSTEM-PI108-02

```

5800      " " " " CUT < CUT      PREDICTED CORRECTLY      PREDICTION RATIO";
5900      FORMAT HOR41AG,AG,"ZONE",PANEL,"X11","MEAN",X8,"STN DEV",X9,"SAMPLE";
6000      FORMAT DET1(K24,3(14,X4),X15,3(14,X4),X8,14,X19,F5,3);
6100      FORMAT DET4(X23,12,X10,F5,1,X10,F5,1,X10,15);
6200      ALPHA BOARD,GRBD;
6300      REAL TVAP,TVAS,TVAT,VAP,VAS,VAT,VARIANT;
6400      REAL TEMP,TEMS,TEMT;
6500      INTEGER BDCNT;
6600      BDCR;
6700      BDCRSO;
6800      BINCNT;
6900      BINCNT1;
7000      CUT;
7100      HOLD1;
7200      HOLD2;
7300      I,M,K,J,L;
7400      NRVAR5,X1,XV,M1,M2,M3;
7500      USM,MRPRL,MSECM,RTIER;
7600      LSPRT,LSSEC,LSTER;
7700      TXI;
7800      PANEL;
7900      SAMP;
8000      SCORE;
8100      SUPP;
8200      VAR;
8300      ZONE;
8400      X;
8500      DEFINE ZONEDEF = INTEGER(BIN141),1),XBINZONE;
8600      BINSEL = BIN1421;
8700      BINPAN = BIN1431;
8800      LABEL READBIN,READBD,ENDRD,COMPARE,PASS1,ENDSTAT;
8900      ENDLOOP,ENDGAIN,ENDGAIN2,CKSCORE;
9000      $ PAGE;
9100      PROCEDURE HICOR(WCHARRY,WCHIA,WCHLS);
9200      ARRAY WCHARRY(*),WCHIA(*);
9300      INTEGER WCHLS;
9400      BEGIN
9500      LABEL OUTHI,ENDM;
9600      ARRAY RIO:1001;
9700      REAL V;
9800      INTEGER MC,VALIA,LL,L,M,J,N;
9900      MC := 0;
10000      LL := (87-1)*NRVAR5;
10100      FOR I := 1 STEP 1 UNTIL NRVAR5 DO
10200      REIJ := WCHARRYLL+1;
10300      FOR M1 := 1 STEP 1 UNTIL WCHLS DO
10400      BEGIN
10500      IF MC = WCHLS THEN GO TO OUTHI;
10600      V := 0;
10700      FOR I := 1 STEP 1 UNTIL WCHLS DO
10800      BEGIN
10900      VALIA := WCHIA(I);
11000      IF ABS(WCHARRY(LL+VALIA)) < V THEN ELSE
11100      BEGIN
11200      V := ABS(WCHARRYLL+VALIA);
11300      L := WCHIA(I);
11400      END;
11500      END;
11600      MC := * + 1;

```

```

11700 WCHARRY(LL+1) := 0.0;
11800 FOR M := 1 STEP 1 UNTIL WCHLS DO
11900 BEGIN
12000 J := WCHIA(M);
12100 IF J = L THEN GO TO ENDM;
12200 IF ABS(WCHARRY((L-1)*NRVARS+J)) < 0.80 THEN ELSE
12300 BEGIN
12400 MC := * + 1;
12500 WRITE(LINE, "VAR "12," REMOVED DUE TO HI CORRELATION",
12600 " WITH VAR "122," J, L);
12700 FOR N := 1 STEP 1 UNTIL NRVARS DO
12800 IF N = J THEN ELSE
12900 BEGIN
13000 RLJ := 0.0017;
13100 WCHARRY((N-1)*NRVARS+J) := 0.0017;
13200 WCHARRY((J-1)*NRVARS+N) := 0.0017;
13300 END;
13400 ENDM; END;
13500
13600 OUTHI: FOR I := 1 STEP 1 UNTIL NRVARS DO
13700 BEGIN
13800 WCHARRY(LL+1) := R(I);
13900 END HICOR;
14000 $ PAGE
14100 PROCEDURE CORPRINT(WCHARRY);
14200 ARRAY WCHARRY(*);
14300 BEGIN
14400 INTEGER J, L, LL, LINCNT;
14500 LINCNT := 0;
14600 FOR L := 1 STEP 12 UNTIL 85 DO
14700 BEGIN
14800 WRITE(LINE, <X9, 12(X4, "VAR"12, X1) />, FOR LL := L STEP 1 UNTIL L+11
14900 DO LL);
15000 FOR J := 1 STEP 1 UNTIL NRVARS DO
15100 BEGIN
15200 WRITE(LINE, <X2, "VAR", 12, X1, 12(F10 5) />, J, FOR LL := L STEP 1
15300 UNTIL L+11 DO WCHARRY((J-1)*NRVARS+LL));
15400 LINCNT := * + 1;
15500 IF LINCNT > 50 THEN
15600 BEGIN
15700 WRITE(LINE(SKIP 1));
15800 LINCNT := 0;
15900 WRITE(LINE, <X9, 12(X4, "VAR"12, X1) />, FOR LL := L STEP 1
16000 UNTIL L+11 DO LL);
16100 END;
16200 END;
16300 WRITE(LINE(SPACE 2));
16400 LINCNT := * + 3;
16500
16600 ENDM;
16700 WRITE(LINE(SKIP 1));
16800 END CORPRINT;
16900 $*****START PROGRAM*****
17000 NRVARS := 96;
17100 BINARY OPEN := TRUE;
17200 B := BINARY MAXREC SIZE;
17300 READ(CRD, CARDIN, BOARD, GRBD, CUTSCORE(2), CUTSCORE(1));
17400 VAR := IF BOARD = "R" THEN 1 ELSE
17500 IF BOARD = "N" THEN 2 ELSE
17600 IF BOARD = "P" THEN 3 ELSE

```



```

17700      4:
17800      READ(BOARDFILE,640,BF);
17900      BDCNT := * + 1;
18000      READBIN;
18100      READ(BINARY,B,BIN)(ENDRD);
18200      BINCNT := * + 1;
18300      IF BINCNT > 1200 THEN GO TO ENDRD;
18400      COMPARE;
18500      IF BF(1) > BIN(0) FOR 9 THEN GO TO READBIN;
18600      IF BF(1) < BIN(0) FOR 9 THEN GO TO READBD;
18700      IF BF(42) = "R" THEN GO TO READBIN;
18800      IF BF(58) > "99" OR BF(58) < "01" THEN GO TO READBIN;
18900      IF BF(59) = "000" THEN GO TO READBIN;
19000      ZONECNT(ZONEDEF) := * + 1;
19100      GO TO PASS1;
19200      READBD;
19300      READ(BOARDFILE,640,BF)(ENDRD:PARR);
19400      BDCNT := * + 1;
19500      GO TO COMPARE;
19600      PASS1: SCORE := INTEGER(BF(37),3) - 300;
19700      BDCNT := INTEGER(BF(58),3);
19800      IF BF(37) > "500" OR BF(37) < "300"
19900      THEN GO TO CKSCORE;
20000      CUT := IF BF(37) > POINT(CUTSCORE(ZONEDEF)) + 3 FOR 3 THEN 0
20100      ELSE IF BF(37) > POINT(CUTSCORE(ZONEDEF)) + 3 FOR 3 THEN 1
20200      ELSE 2;
20300      X0-PRED-CUT,1-PRED-CUT,2-PRED-CUT
20400      CASE VAR OF
20500      STATDIST(ZONEDEF,SCORE,0) := * + 1;
20600      BEGIN
20700      1: IF BF(249) = "N" THEN
20800      BEGIN
20900      REPLACE BINSEL BY "X";
21000      IF X1 := 0;
21100      END ELSE
21200      IF X1 := 1;
21300      IF BF(158) = "3" THEN
21400      BEGIN
21500      REPLACE BINSEL BY "X";
21600      IF X1 := 0;
21700      END ELSE
21800      STATDIST(ZONEDEF,SCORE,1) := * + 1;
21900      END ELSE
22000      IF BF(156) = "X" THEN
22100      BEGIN
22200      REPLACE BINSEL BY "X";
22300      IF X1 := 0;
22400      END ELSE
22500      STATDIST(ZONEDEF,SCORE,1) := * + 1;
22600      END ELSE
22700      IF BF(157) = "X" THEN
22800      BEGIN
22900      REPLACE BINSEL BY "X";
23000      IF X1 := 0;
23100      END ELSE
23200      STATDIST(ZONEDEF,SCORE,1) := * + 1;
23300      END ELSE
23400      IF X1 := 1;
23500      END;
23600      STATDIST(ZONEDEF,IX1,CUT) := * + 1;

```

```

23700 CKSCORE:
23800 REPLACE BINPAN BY BF(581) FOR 2;
23900 REPLACE BIN(124) BY BF(593) FOR 3;
24000 BDCRSQ := BDCSR = BDCSR;
24100 DISTZONEDEF, INTEGER(BF(581), 2), 0) := # + BDCSR;
24200 DISTZONEDEF, 100, 0) := # + BDCSR;
24300 DISTZONEDEF, INTEGER(BF(581), 2), 1) := # + BDCRSQ;
24400 DISTZONEDEF, 100, 1) := # + BDCRSQ;
24500 DISTZONEDEF, INTEGER(BF(581), 2), 2) := # + 1;
24600 DISTZONEDEF, 100, 2) := # + 1;
24700 WRITE(BINARY, B, BIN);
24800 GO TO READBIN;
24900
25000 PARR:
25100 PARCNT := # + 1; DOO := READ(BOARDFILE, 640, BF); GO TO READBIN;
25200
25300 ENDRO:
25400 REVIND(BINARY);
25500 WRITE(LINE, SPACE 21, <"PREPARED(MMOYV)"; "AG", TIME(15));
25600 WRITE(LINE, SPACE 21, <"BOARD"; "A1, A3", BOARD, GRBD);
25700 FOR X := 1 STEP 1 UNTIL 3 DO
25800 BEGIN
25900 IF ZONECNT(X) = 0 THEN GO TO ENDSTAT;
26000 WRITE(LINE, <AG, AG>, TITLE(X, 12), TITLE(X, 12+6));
26100 WRITE(LINE, SPACE 21, HDR3);
26200 HOLD1 := HOLD2 := 0;
26300 FOR I(X) := 0, 1 DO
26400 FOR CUT := 0 STEP 1 UNTIL 2 DO
26500 HOLD2 := # + STATDIST2(X, I, CUT);
26600 HOLD1 := STATDIST2(X, 0, 0) + STATDIST2(X, 1, 2) +
26700 STATDIST2(X, 0, 1);
26800 WRITE(LINE, SPACE 31, DET1, STATDIST2(X, 0, 0),
26900 STATDIST2(X, 1, 0), STATDIST2(X, 1, 1),
27000 STATDIST2(X, 1, 2), HOLD1, HOLD1/HOLD2);
27100 ENDSTAT: END;
27200 CASE VAR OF
27300 BEGIN
27400 WRITE(LINE, HDR1);
27500 WRITE(LINE, SPACE 21, HDR2R);
27600 FOR SCORE := 0 STEP 1 UNTIL 200 DO
27700 WRITE(LINE, <X4, X9, X9, X4, X7, X7, X7>, SCORE+300,
27800 STATDIST1(1, SCORE, 0), STATDIST1(1, SCORE, 1),
27900 STATDIST1(2, SCORE, 0), STATDIST1(2, SCORE, 1),
28000 STATDIST1(3, SCORE, 0), STATDIST1(3, SCORE, 1));
28100 WRITE(LINE, HDR1);
28200 WRITE(LINE, SPACE 21, HDR2N);
28300 FOR SCORE := 0 STEP 1 UNTIL 200 DO
28400 WRITE(LINE, <X4, X13, X13, X14, X7, X7, X7>, SCORE+300,
28500 STATDIST1(1, SCORE, 0), STATDIST1(1, SCORE, 1));
28600 WRITE(LINE, HDR1);
28700 WRITE(LINE, SPACE 21, HDR2N);
28800 FOR SCORE := 0 STEP 1 UNTIL 200 DO
28900 WRITE(LINE, <X4, X13, X13, X14, X7, X7, X7>, SCORE+300,
29000 STATDIST1(1, SCORE, 0), STATDIST1(1, SCORE, 1));
29100 WRITE(LINE, HDR1);
29200 WRITE(LINE, SPACE 21, HDR2T);
29300 FOR SCORE := 0 STEP 1 UNTIL 200 DO
29400 WRITE(LINE, <X4, X9, X9, X4, X7, X7, X7>, SCORE+300,
29500 STATDIST1(1, SCORE, 0), STATDIST1(1, SCORE, 1),
29600 STATDIST1(2, SCORE, 0), STATDIST1(2, SCORE, 1));

```

```

28700      END;
28800      FOR ZONE := 1 STEP 1 UNTIL 3 DO
28900          BEGIN
30000              IF ZONECNT(ZONE) = 0 THEN GO TO ENDOOP;
30100              FOR PANEL := 1 STEP 1 UNTIL 100 DO
30200                  BEGIN
30300                      SAMPL := DIST(ZONE,PANEL,2);
30400                      SUMM := DIST(ZONE,PANEL,0);
30500                      IF SAMPL > 0 THEN
30600                          BEGIN
30700                              HOLD1 := (SAMPL * DIST(ZONE,PANEL,1)) -
30800                                  (SUMM * SUMM);
30900                              HOLD2 := HOLD1/(SAMPL * SAMPL);
31000                              DIST(ZONE,PANEL,1) := SORT(HOLD2);
31100                              DIST(ZONE,PANEL,0) := SUMM/SAMPL;
31200                              END;
31300                          END;
31400                      ENDOOP;
31500                      READ(BINARY,B,BIN)(ENDAGAIN);
31600                      BINCNT1 := * + 1;
31700                      IF BINCNT1 > 1200 THEN GO TO ENDAGAIN;
31800                      IF BINPAN > "99" OR BINPAN < "01" THEN GO TO READAGAIN;
31900                      IF BIN(124) = "000" THEN GO TO READAGAIN;
32000                      REPLACE BIN(127) BY
32100                          INTEGER(BIN(124),3) - ((DIST(ZONEDEF,INTEGER(BINPAN,2),0)
32200                              - DIST(ZONEDEF,100,0))) FOR 3 DIGITS;
32300                      WRITE(BINARY,B,BIN);
32400                      *****THIS IS THE NEW
32500                      ZONEDEF := * + 1;
32600                      FOR X := 1 STEP 1 UNTIL 100 DO RECHOLD(X) := 0;
32700                      X := 1;
32800                      FOR I := 32 STEP 1 UNTIL 40,41,43,45 STEP 1 UNTIL 98,99 STEP 2
32900                          UNTIL 107,109 STEP 1 UNTIL 123,124,127,130 STEP 1 UNTIL 134,
33000                          135,137,139,140 DO
33100                              BEGIN
33200                                  VARANT := IF I=41 OR I=43 THEN INTEGER(BIN(1),2) ELSE
33300                                      IF I > 98 AND I < 108 THEN INTEGER(BIN(1),2)/10 ELSE
33400                                      IF I=124 OR I=127 THEN INTEGER(BIN(1),3) ELSE
33500                                      IF I=135 OR I=137 THEN INTEGER(BIN(1),2)/10 ELSE
33600                                      IF I=139 THEN INTEGER(BIN(1),1);
33700                                  RECHOLD(X) := VARANT;
33800                                  IF VARANT > 0 THEN
33900                                      BEGIN
34000                                          IF ZONEDEF = 1 THEN
34100                                              BEGIN
34200                                                  SUMPT(X) := * + VARANT;
34300                                                  SUMPS(X) := * + VARANT**2;
34400                                                  END ELSE
34500                                                  IF ZONEDEF = 2 THEN
34600                                                      BEGIN
34700                                                          SUMS(X) := * + VARANT;
34800                                                          SUMSS(X) := * + VARANT**2;
34900                                                          END ELSE
35000                                                          IF ZONEDEF = 3 THEN
35100                                                              BEGIN
35200                                                                  SUMT(X) := * + VARANT;
35300                                                                  SUMTS(X) := * + VARANT**2;
35400                                                                  END;
35500                                                              END;
35600                                  END;

```



```

35700      X := # + 1;
35800      END;
35900      FOR X := 1 STEP 1 UNTIL NRVAR#-1 DO
36000          BEGIN
36100              VARIANT := RECHOLD(X+1);
36200              IF VARIANT = 0 THEN ELSE
36300                  BEGIN
36400                      M := X*NRVAR#;
36500                      FOR K := 1 STEP 1 UNTIL X DO
36600                          IF RECHOLD(K) > 0 THEN
36700                              BEGIN
36800                                  IF ZONEDEF = 1 THEN
36900                                      XYSUMPT(M+K) := # + (VARIANT*RECHOLD(K)) ELSE
37000                                      IF ZONEDEF = 2 THEN
37100                                          XYSUMS(M+K) := # + (VARIANT*RECHOLD(K)) ELSE
37200                                          IF ZONEDEF = 3 THEN
37300                                              XYSUMT(M+K) := # + (VARIANT*RECHOLD(K));
37400                                  END;
37500                              END;
37600                          END;
37700                      GO TO READAGAIN;
37800                      ENDAGAIN;
37900                      FOR X := 1 STEP 1 UNTIL 3 DO
38000                          BEGIN
38100                              IF ZONECNT(X) = 0 THEN GO TO ENLOOP2;
38200                              WRITE(LINE[SKIP,1]);
38300                              WRITE(LINE[SPACE,2],H0R4,TITL(X*12),TITL(X*12+6));
38400                              FOR PANEL := 1 STEP 1 UNTIL 100 DO
38500                                  IF DIST(X,PANEL,2) > 0 THEN
38600                                      WRITE(LINE[DETA,PANEL,DIST(X,PANEL,0)],
38700                                          DIST(X,PANEL,1),DIST(X,PANEL,2));
38800                                  ENLOOP2: END;
38900                                  X*****NEW
39000                                  XX := NRVAR# DIV 8; XY := NRVAR# MOD 8;
39100                                  IF ZONECNT(1) > 0 THEN
39200                                      BEGIN
39300                                          M1 := ZONECNT(1) - 1;
39400                                          FOR X := 1 STEP 1 UNTIL NRVAR# DO
39500                                              BEGIN
39600                                                  REAMP(X) := SUMPT(X)/ZONECNT(1);
39700                                                  STDVPT(X) := (SUMPT(X)-ZONECNT(1)*(REAMP(X)+2))/M1;
39800                                                  IF STDVPT(X) > 0 THEN STDVPT(X) := STDVPT(X)*0.5;
39900                                                  END;
40000                                                  FOR X := 0 STEP 1 UNTIL XX - 1 DO
40100                                                      WRITE(PRI,<8F10.5,X2>); FOR K := 1 STEP 1 UNTIL 8 DO
40200                                                          MEAMP(X+8*K);
40300                                                      IF XY > 0 THEN
40400                                                          MEAMP(X+8*K);
40500                                                      FOR X := 0 STEP 1 UNTIL XX - 1 DO
40600                                                          WRITE(PRI,<8F10.5,X2>); FOR K := 1 STEP 1 UNTIL 8 DO
40700                                                              STDVPT(X+8*K);
40800                                                              IF XY > 0 THEN
40900                                                                  STDVPT(X+8*K);
41000                                                                  WRITE(PRI,<8F10.5,X2>); FOR K := 1 STEP 1 UNTIL XY DO
41100                                                                      STDVPT(X+8*K);
41200                                                                  END;
41300                                                                  IF ZONECNT(2) > 0 THEN
41400                                                                      BEGIN
41500                                                                          M2 := ZONECNT(2) - 1;
41600                                                                          FOR X := 1 STEP 1 UNTIL NRVAR# DO

```



```

41700      BEGIN
41800      MEANS(X) := SUMS(X)/ZNCNT(2);
41900      STDS(X) := ((SUMS(X)-ZNCNT(2)*(MEANS(X)*2))/M2);
42000      IF STDS(X) > 0 THEN STDS(X) := STDS(X)*0.5;
42100      END;
42200      **WRITE MEANS
42300      FOR X := 0 STEP 1 UNTIL XX - 1 DO
42400      WRITE(SEC,<8F10.5,X2>,<FOR K := 1 STEP 1 UNTIL 8 DO
42500      MEANS(X*8+K));
42600      IF XY > 0 THEN
42700      WRITE(SEC,<8F10.5,X2>,<FOR K := 1 STEP 1 UNTIL XY DO
42800      MEANS(X*8+K));
42900      FOR X := 0 STEP 1 UNTIL XX - 1 DO
43000      WRITE(SEC,<8F10.5,X2>,<FOR K := 1 STEP 1 UNTIL 8 DO
43100      STDS(X*8+K));
43200      IF XY > 0 THEN
43300      WRITE(SEC,<8F10.5,X2>,<FOR K := 1 STEP 1 UNTIL XY DO
43400      STDS(X*8+K));
43500      END;
43600      IF ZNCNT(3) > 0 THEN
43700      BEGIN
43800      M3 := ZNCNT(3) - 1;
43900      FOR X := 1 STEP 1 UNTIL NRVAR5 DO
44000      BEGIN
44100      MEANT(X) := SUMT(X)/ZNCNT(3);
44200      STDV(X) := ((SUMT(X)-ZNCNT(3)*(MEANT(X)*2))/M3);
44300      IF STDV(X) > 0 THEN STDV(X) := STDV(X)*0.5;
44400      END;
44500      FOR X := 0 STEP 1 UNTIL XX - 1 DO
44600      WRITE(TER,<8F10.5,X2>,<FOR K := 1 STEP 1 UNTIL 8 DO
44700      MEANT(X*8+K));
44800      IF XY > 0 THEN
44900      WRITE(TER,<8F10.5,X2>,<FOR K := 1 STEP 1 UNTIL XY DO
45000      MEANT(X*8+K));
45100      FOR X := 0 STEP 1 UNTIL XX - 1 DO
45200      WRITE(TER,<8F10.5,X2>,<FOR K := 1 STEP 1 UNTIL 8 DO
45300      STDV(X*8+K));
45400      IF XY > 0 THEN
45500      WRITE(TER,<8F10.5,X2>,<FOR K := 1 STEP 1 UNTIL XY DO
45600      STDV(X*8+K));
45700      END;
45800      FOR I := 2 STEP 1 UNTIL NRVAR5 DO
45900      BEGIN
46000      L := (I-1) * NRVAR5;
46100      TEMP := SUMP(I); TEMS := SUMS(I); TENT := SUMT(I);
46200      IF ZNCNT(I) > 0 THEN
46300      TVAP := STDV(I);
46400      IF ZNCNT(2) > 0 THEN
46500      TVAS := STDS(I);
46600      IF ZNCNT(3) > 0 THEN
46700      TVAT := STDV(I);
46800      FOR J := 1 STEP 1 UNTIL I-1 DO
46900      BEGIN
47000      M := L+J;
47100      IF ZNCNT(I) > 0 THEN
47200      BEGIN
47300      VAP := TVAP*STDV(J);
47400      IF VAP > 0 THEN
47500      BEGIN
47600      XVSUMP(M) := ((XVSUMP(M) - (TEMP*SUMP(J)/ZNCNT(I)))/

```

```

47700 M1)/VAP;
47800 IF ABS(XYSUMPM(J)) < 0.80 THEN ELSE
47900 IF I = 86 OR I = 87 THEN ELSE
48000 IF J = 86 OR J = 87 THEN ELSE
48100 BEGIN
48200 ISW := 0;
48300 FOR MRPRI := 1 STEP 1 UNTIL LSPRI+1 DO
48400 IF IAPRI(MRPRI) = 1 THEN ISW := 1;
48500 IF ISW = 1 THEN ELSE
48600 BEGIN
48700 LSPRI := # + 1;
48800 IAPRI(LSPRI) := 1;
48900 END;
49000 ISW := 0;
49100 FOR MRPRI := 1 STEP 1 UNTIL LSPRI+1 DO
49200 IF IAPRI(MRPRI) = 1 THEN ISW := 1;
49300 IF ISW = 1 THEN ELSE
49400 BEGIN
49500 LSPRI := # + 1;
49600 IAPRI(LSPRI) := 1;
49700 END;
49800
49900 END ELSE XYSUMPM(M) := 0.0019;
50000 XYSUMPM((J-1)*NRVARS)+1 := XYSUMPM(M);
50100 END;
50200 IF ZNCNT(2) > 0 THEN
50300 BEGIN
50400 VAS := TVAS+STDV(J);
50500 IF VAS > 0 THEN
50600 BEGIN
50700 XYSUMPM := ((XYSUMPM(M)-(TENS*SUMS(J)/ZNCNT(2)))/
50800 M2)/VAS;
50900 IF ABS(XYSUMPM(M)) < 0.80 THEN ELSE
51000 IF I = 86 OR I = 87 THEN ELSE
51100 IF J = 86 OR J = 87 THEN ELSE
51200 BEGIN
51300 ISW := 0;
51400 FOR MRSEC := 1 STEP 1 UNTIL LSSEC+1 DO
51500 IF IASEC(MRSEC) = 1 THEN ISW := 1;
51600 IF ISW = 1 THEN ELSE
51700 BEGIN
51800 LSSEC := # + 1;
51900 IASEC(LSSEC) := 1;
52000 END;
52100 ISW := 0;
52200 FOR MRSEC := 1 STEP 1 UNTIL LSSEC+1 DO
52300 IF IASEC(MRSEC) = 1 THEN ISW := 1;
52400 IF ISW = 1 THEN ELSE
52500 BEGIN
52600 LSSEC := # + 1;
52700 IASEC(LSSEC) := 1;
52800 END;
52900
53000 END ELSE XYSUMPM(M) := 0.0019;
53100 XYSUMPM((J-1)*NRVARS)+1 := XYSUMPM(M);
53200 END;
53300 IF ZNCNT(3) > 0 THEN
53400 BEGIN
53500 VAT := TVAT+STDV(J);
53600 IF VAT > 0 THEN

```

```

53700 BEGIN
53800 XYSUMT[M] := ((XYSUMT[M] - (TEMP*SUMT[LJ]/ZNCNT[LJ]))/
53900 R3)/VAT;
54000 IF ABS(XYSUMT[M]) < 0.80 THEN ELSE
54100 IF I = 86 OR I = 87 THEN ELSE XDEP VARIABLE
54200 IF J = 86 OR J = 87 THEN ELSE
54300 BEGIN
54400 ISW := 0;
54500 FOR MTR := 1 STEP 1 UNTIL LSTER+1 DO
54600 IF IATER[MTR] = 1 THEN ISW := 1;
54700 IF ISW = 1 THEN ELSE
54800 BEGIN
54900 LSTER := * + 1;
55000 IATER[LSTER] := 1;
55100 END;
55200 ISW := 0;
55300 FOR MTR := 1 STEP 1 UNTIL LSTER+1 DO
55400 IF IATER[MTR] = J THEN ISW := 1;
55500 IF ISW = 1 THEN ELSE
55600 BEGIN
55700 LSTER := * + 1;
55800 IATER[LSTER] := J;
55900 END;
56000 END ELSE XYSUMT[M] := 0.0019;
56100 XYSUMT[LJ-1+NRVARS+1] := XYSUMT[M];
56200 END;
56300 XYSUMT[L+1] := XYSUMS[L+1] := XYSUMT[L+1] := 1.0;
56400 END;
56500 XYSUMP[L+1] := XYSUMS[L+1] := XYSUMT[L+1] := 1.0;
56600 END;
56700 XYSUMP[L] := XYSUMS[L] := XYSUMT[L] := 1.0;
56800 I := NRVARS+2;
56900 XYSUMP[L] := XYSUMS[L] := XYSUMT[L] := 1.0;
57000 IF ZNCNT[L] > 0 THEN
57100 BEGIN
57200 WRITE(LINE(SKIP 1));
57300 WRITE(LINE, "PRIMARY/PILOT ZONE CORRELATION MATRIX"/>);
57400 CORPRINT(XYSUMP);
57500 WRITE(LINE, "PRIMARY/PILOT ZONE HICOR"/>);
57600 HICOR(XYSUMP, IAPRI, LSPRI);
57700 END;
57800 IF ZNCNT[L2] > 0 THEN
57900 BEGIN
58000 WRITE(LINE(SKIP 1));
58100 WRITE(LINE, "SECONDARY/NAV ZONE CORRELATION MATRIX"/>);
58200 CORPRINT(XYSUMS);
58300 WRITE(LINE, "SECONDARY/NAV ZONE HICOR"/>);
58400 HICOR(XYSUMS, IASEC, LSECC);
58500 END;
58600 IF ZNCNT[L3] > 0 THEN
58700 BEGIN
58800 WRITE(LINE(SKIP 1));
58900 WRITE(LINE, "TERTIARY/SPT ZONE CORRELATION MATRIX"/>);
59000 CORPRINT(XYSUMT);
59100 WRITE(LINE, "TERTIARY/SPT ZONE HICOR"/>);
59200 HICOR(XYSUMT, IATER, LSTER);
59300 END;
59400 X* = NRVARS DIV 8, XY = NRVARS MOD 8 (CONTROL NR LINES WRITTEN)
59500 FOR I := 0 STEP 1 UNTIL NRVARS-1 DO
59600 BEGIN

```



```

59700 L := 1 + NRVARS;
59800 IF ZNCNT(1) > 0 THEN
59900 BEGIN
60000 FOR X := 0 STEP 1 UNTIL XX-1 DO
60100 WRITE(PRI, <F10.5,X2>, FOR K := 1 STEP 1 UNTIL 8 DO
60200 XYSUMFIL+(X+8*K));
60300 IF XY > 0 THEN
60400 WRITE(PRI, <F10.5,X2>, FOR K := 1 STEP 1 UNTIL XY DO
60500 XYSUMFIL+(X+8*K));
60600 END;
60700 IF ZNCNT(2) > 0 THEN
60800 BEGIN
60900 FOR X := 0 STEP 1 UNTIL XX-1 DO
61000 WRITE(SEC, <F10.5,X2>, FOR K := 1 STEP 1 UNTIL 8 DO
61100 XYSUMS(L+(X+8*K));
61200 IF XY > 0 THEN
61300 WRITE(SEC, <F10.5,X2>, FOR K := 1 STEP 1 UNTIL XY DO
61400 XYSUMS(L+(X+8*K));
61500 END;
61600 IF ZNCNT(3) > 0 THEN
61700 BEGIN
61800 FOR X := 0 STEP 1 UNTIL XX-1 DO
61900 WRITE(TER, <F10.5,X2>, FOR K := 1 STEP 1 UNTIL 8 DO
62000 XYSUMTIL+(X+8*K));
62100 IF XY > 0 THEN
62200 WRITE(TER, <F10.5,X2>, FOR K := 1 STEP 1 UNTIL XY DO
62300 XYSUMTIL+(X+8*K));
62400 END;
62500 IF ZNCNT(1) > 0 THEN
62600 LOCK(PRI);
62700 IF ZNCNT(2) > 0 THEN
62800 LOCK(SEC);
62900 IF ZNCNT(3) > 0 THEN
63000 LOCK(TER);
63100 WRITE(LINE(SKIP 1));
63200 WRITE(LINE(SKIP 1), <K61>, "END REPORT");
63300 WRITE(LINE, <F10.5,X2>, "A62 TIME(15)");
63400 WRITE(LINE, <F10.5,X2>, "A1 A3 BOARD GRD");
63500 WRITE(LINE, <X10>, "BOARD COUNT - ", 16> BNCNT);
63600 WRITE(LINE, <X10>, "BIN REC IN - ", 16> BNCNT);
63700 WRITE(LINE, <X10>, "BIN REC IN/OUT - ", 16> BNCNT);
63800 WRITE(LINE, <X10>, "RECORD COUNTS INPUT TO REG BUILD - ",
63900 315, X2>, FOR I := 1 STEP 1 UNTIL 3 DO ZNCNT(1));
64000 WRITE(LINE, <X10>, "PROCESS TIME - ", F6.2, " SECS", TIME(2)/60);
64100 WRITE(LINE, <X10>, "I/O TIME - ", F6.2, " SECS", TIME(3)/60);
64200 WRITE(LINE, <X10>, "PARITY ERR CNT - ", 13> PARCNT);
64300 END;
64400

```